ril lberga Ro dler Richard nderson
eorge ndrews Henr Baker te hen Bal ac
Yurij Baransk David R. Barton erald Baumgartner
ilberR.RoRo R. roDavid-R.Da ak Tej32.560Tk a Tj 6. 20 Mumga Tj5.040Tn k Tj29.680Tuel er

ntent

New Foreword

In October , 200 Axiom was withdrawn from the market and ended life as a commercial product.

Chapter 1

Axiom Features

1.1 Introduction to Axiom

Welcome to the world of Axiom. We call Axiom a

which would g

literally dozens of kinds of numbers to compute with. These range from various kinds of in

```
nverse(%)
```

$$\begin{bmatrix} \frac{1}{x\ \ i} & 0 \\ \frac{1}{2\ x\ \ 2\ i} & -\frac{1}{2} \end{bmatrix}$$

```
      e    n n(Matr x Fract n P  n  a C   ex nte er,   )
```

1 1 4 H erDo

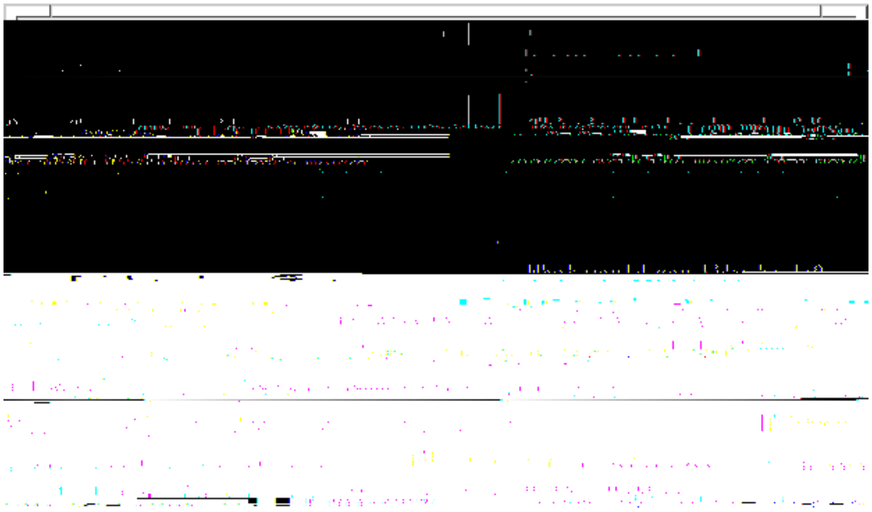

Figure . H erdoc o ening menu

H erDoc resents ou windows o

k

```
draw(5 besse J(0,s rt(x  2    2)), x -20  20,    -20  20)
```

Figure .2 J_0

$$\Bigg[\ ,3\ x,\ ^{5}$$

```
                 e   Ex ress  n  nte er
```

Note the use of %” here. This means the value of the last expression we computed. In this case it is the long expression above.

1 1 8 P ttern t h 8 T 106P8 T 4 320T9T T 10 080TdtheT e

Using **input** files and the **read** command, you can create your o

2111 XI

Chapter 2

Ten Fundamental Ideas

`e Matrix P n a Fract n nte er`

the interpreter

Axiom's use of abstract datatypes clearly separates the exports of a domain what operations are defined fro

h ter 3

t rting Axi

Welcome to the xiom environment for interactivvir

If you are running Axiom under the X Window System, there may be two windows: the console window (as just described) and the HyperDoc main menu. HyperDoc is a multiple-window hypertext system that lets you view Axiom documentation and examples on-line, execute Axiom expressions, and generate graphics. If you are in a gra

```
1   2 - 3 / 4   3    2 - 1
```

$$-\frac{9}{4}$$

```
                                         ▘ e   Fract ▖n  nte er
```

The above expression is equivalent to this.

```
((1   2) - ((3 / 4)   (3    2))) - 1
```

$$-\frac{9}{4}$$

```
                                         ▘ e   Fract ▖n  nte er
```

If an expression contains subexpressions enclosed in parentheses, the ar

$$9999999999$$

```
■ e P■s t ve nte er
```

This is the last result.

```
%%(-1)
```

$$9999999999$$

```
■ e P■s t ve nte er
```

This is the result from ste number .

```
%%(1)
```

$$0000000000$$

```
■ e P■s t ve nte er
```

3 2 3 So■e T es

ver thing in xiom has a t e. The t e determines what o erations ou can erform on an object and how the ob■e[

egeege

```
■ e P■s t ve nte er ■ 11 s880■d(r )■ 15 6■J18 8a11 640■0(
```

```
x  8
```

$$x^8$$

`e  P  n  a  nte er`

Here a negative integer exponent produces a fraction

This gives the value z $3/5$ a

e F at

Use . 9 rse

3.2.6 Calling Functions

As we saw earlier, when you want to add or subtract two values, you place the arithmetic operator ”tt

n operations that returns a `Boolean` value that is, `true` or `false`

r

3 3 2 T e Conv

3 3 3 Useful Functions

To obtain the absolute value of a x n 6the onsn Tf offoi.5d[tnng[- 3.Td F Tjle.840Tthea T th

e Ps t ve nte er

Tests on values can be done using various functions which are generally more efficient than using relational operators such as particularly if the value is a matrix. Examples of s 2Tfl tmsaTj 3mue

e B ean

e

3.4.2 Complex Numbers

For many scientific calculations real numbers aren't sufficient and support for complex numbers is also required. Complex numbers are handled in an intuitive manner. Axiom uses the **%i** macro to represent the squa%

`e C ex nte er`

`act r(%)`

i

```
rad x(3/21,5)
```

$$0.\overline{0324\ 2}$$

```
                                   ■   e   K412
                                                     ■   41■41
```

`c actFract n(%)`

$6 - {}^{3}$

_ 3

The first example should be read as

`Let` x `be ■ t e PrimeField( ) and assign t■ t the value` 5

Note that it is only possible to invert non-zero values if the arithmetic is performed modulo a prime number. Thus arithmetic modulo a non-prime integer is possible but the reciprocal operation is undefined sn

3 5 4 Comments

file. To get Axiom to read this file, you use the system command `)read input`. If you need to make changes to your approach or definitions, go into your favorite editor, change **input**, then `)read input` again.

Other system commands include `)history`, to display previous input and/or output lines `)display`, to display properties and values of workspace variables and `)what`.

Issue `)what` to get a list of Axiom objects

```
reverse(  ,2,-1,2])
```

[2, − , 2,

```
                                        ■  e   L st  nte er
```

```
s■rt(  ,2,-1,2])
```

[− , 2, 2,

```
                                        ■  e   L st  nte er
```

```
re ■veDu   cates( 1,5,3,5,1,1,2])
```

[, 5, 3, 2

```
                                   ■  e   L st P■s t ve nte er
```

```
#  ,2,-1,2]
```

4

```
                                      ■  e   P■s t ve nte er
```

Lists in xiom are mutable and so their contents the elemen the5 o te -d3/R240. 2Tld[th - 000 e hc ksT ande

$$[9, 2, 4, \ , \ , 5, 42$$

```
■ e L st P■s t ve nte er
```

```
end0 u   rest(u,4)
```

$$[\ , 5, 42$$

```
■ e L st P■s t ve nte er
```

```
 art0 u   rest(u,2)
```

$$[4, \ , \ , 5, 42$$

```
■ e L st P■s t ve nte er
```

```
setrest!(end0 u, art0 u)  u
```

$$[9, 2,$$

$$[9, 99, 20,$$

e L st P s t ve nte er

In the previous example a new

3 6 2 Se ented L sts

segmented list is one in which some of the elements are ranges of values. The **x and** function converts lists of this t e into ordinar lists

```
1  10]
```

[.. 0

```
 e  L st
```

To create the series the window is laced at the star0

`x`

```
swa !(b,2,3)  b
```

$$[2, 4, 3, 5, 6$$

■ e OneD ens ■na Arra P■s t ve nte er

```
c■   nt■!(a,b,3)
```

$$[4, 4, 2 \ l \ rr4, 2, 4, 2 \quad , 6$$

■ e OneD ens ■na Arra

3.6.5 Flexible Arrays

Flexible arrays are designed to provide the efficiency of one-dimensional arrays while r r

```
                                        ■  e   F ex b eArra   nte er
```

```
de ete!( ,5)
```

$$[4, 3, 42, 8, 2, 28$$

```
                                        ■  e   F ex b eArra   nte er
```

```
    (3  5)
```

$$[42, 8, 2$$

```
                                        ■  e   F ex b eArra   nte er
```

```
  2
```

$$[4, 3, 42, 8, 2, 28$$

```
                                        ■  e   F ex b eArra   nte er
```

```
          c  a   b
          c
    )
```

2.82842 24 46 9009 6

```
                                                e    F  at
```

Note that indentation is **extremely** important. If the example abov

```
Error  A  M ss n   ate
L ne   2          a  3 0
L ne   3          b  1 0
L ne   4          c  a   b
L ne   5          c
L ne   6          )
                 A
Error  A  ( ro  A u  to A)   nored
Error  A     ro er s ntax
Error  A  s ntax error at to   eve
Error  A  Poss b     ss n  a )
 5 error(s)  ars n
```

a similar error will be raised. Finall , the ” m

rot

```
3.0
```

e F at

```
b  1 0
```

```
.0
```

e F at

```
c  a  b
```

```
4.0
```

e F at

```
s rt(4 0   c)
```

```
2.82842  24  46 9009 6
```

e F at

which achieves the same result and is easier to understa

with some invocations of these functions

```
 ()

   C▪     n   unct ▪n   w t  t  e () -  L st  nte er

 (4)

   C▪     n   unct ▪n   w
```

[redacted]

```
 (2,9)

   C▪     n   unct ▪n   w

   e   L st  nte er
```

nte er - nte er

e V d

x (a

```
   1
re eat
        4 t en brea
  ■ut ut( )
      1
```

the **r ad** ields

```
   1
```

■ e P■s t ve nte er

```
re eat
        4 t en brea
  ■ut ut( )
      1

   1
   2
   3
   4
```

■ e V■ d

It was mentionsnm00TidTR333B0Tt5n4028. T .44 TJ 2. 20T62Tda4 TJ 2k 202Tf 64. 6248Td[m - o0

0

e N nNe at ve nte er

re eat
1
6 t en brea

`■ d(■)■ 40■de G`

4

e Ps t ve nte er

```
r    1
```

e Ps t ve nte er

```
w   e r <   astrw re eat
  c    1 --  ndex     rst c u n
  w   e c <   astc  re eat
       e t( ,r,c) < 0 t en
      ut ut  r,c,e t( ,r,c)]
      r     astrw
      brea  --
```

e

the **r ad** ields

```
 r a  n 1  4  r b  n 8  5 b  -1 re eat
  ut ut  a,b]

    1,8]
    2, ]
    3,6]
    4,5]
```

```
                                            e  V d
```

Note that without the b - " the segment 8..5T "
NTb ..5 T8 TJ 4.4

■ e Fact■red nte er

Integers can also be displayed to bases other than 0. This is an integer in base .

```
rad x(2593 424601,11)
```

```
0000000000
```

■ e Rad xEx ans ■n 11

Roman numerals arTj2 n2mmera

```
                                            Type: SingleInteger
```

Machine double-precision floating-point numbers are also available for numeric and graphical applications.

```
123.21@DoubleFloat
```

$$23.2\ 00000000000$$

```
                                            Type: DoubleFloat
```

The normal floating-point type in Axiom, `Float`, is a software implementation of floating-point numbers in which the exponent and the mantissa may have an

```
d   ts(40)  ex (%      s rt 163 0)
```

26253 4 2640 68 43.9999999999 992500 259 6

```
e   F at
```

Here are com lex numbers with rational numbers 40Fl%/R8 03950Here880 Here Tj23.520

$$u \quad v\, i$$

```
■ e C■ ex P■ n■ a nte er
```

Of course, you can do complex arithmetic with these also.

```
%   2
```

$$-v^2 \quad u^2 \quad 2\, u\, v\, i$$

```
■ e C■ ex P■ n■ a nte er
```

ver rational number haor6Tf 0526. 6T8ehaor6Tf 05 r int legeege

ince is rime, ou can invert nonzero values.

```
1/x
```

3

```
■ e Pr eF e d
```

You can also compute modulo an integer that is not a rime.

```
    nte erM■d 6    5
```

5

```
■ e    nte erM■d 6
```

ll of the usual arithmetic operations are available.

```
  3
```

5

```
■ e    nte erM■d 6
```

Inversion is noTd1n5oTj/R2f2d0efTj4in5ejd2 erTd 5 Tj/R8 0. 2

Pte gbkfrjn5. 2a0Td P Tj5. 6n0Td od Tj 5. 2 3Td od Tj 5. 2u0Td P Tj5. 6nxTd od Tj 5. 2 o

This defines to be an algebraic number, that is, a aa480Td ebra Tj5480Td ebra Tj 80Td4 a Tj8 a48

`2/% 1`

$$\frac{\begin{gathered}\Big({}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad) \, b^3 \qquad {}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad) \, b^2 \\ {}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad) \, b \quad {}^4 - {}^3 \;\; 2 \;\; {}^2 - \quad 3 \qquad \Big)\end{gathered}}{\begin{gathered}\Big({}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad) \, b^3 \qquad {}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad) \, b^2 \\ {}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad) \, b \quad {}^4 - {}^3 \;\; 2 \;\; {}^2 - \qquad \Big)\end{gathered}}$$

`e   Ex ress n  nte er`

But we need to rationa

```
u      1,-  ,11]
```

$$[\ ,-\ ,\overline{\quad\ ,9}$$

```
                                                   e   L st  nte er
```

A *stream* is a structure that potentially has an infinite number of distinct elements. Think of a stream as an "infinite list" where elements are computed successively.

Create an infinite stream of factored integers. Only a certain numb

ne-dimensional arrays are also mutable ou can change their constituent elements in place.”

```
a 3    11  a
```

$$\left[\ ,- \ , \quad , \frac{3}{2} \right]$$

```
                      ■  e   OneD  ens ■na Arra  Fract ■n  nte er
```

However, one-dimensional arrays are not flexible structures. You cannot destructively **concat!** them together.

```
c■ncat!(a,■neD  ens ■na Arra   1,-2])
```

```
   ■ ere are 5 ex ■sed and 0 unex ■sed   brar  ■ erat ■ns na ed c■ncat!
       av n 0c[illegible]
```

ibl *rr*ɐ λp ɐʇ

`■ e Mu t set nte er`

A *table* is conceptually a set of "key–value" pairs and is a generalization of a multiset. For examples of tables, s

```
dan e     Rec■rd(a e     nte er, sa ar
```

3.10 Expanding to Higher Dimensions

To get higher dimensional aggregates,

numbers as coefficients. Moreover, the librar rovides a w

This function is lessft thanthe version sincefcinterationin-volvfarecursivefunctio

```
ac(
```

e Pus t ve nte er

The librar version uses an algorithm that is differen

reate an exam le matrix to ermute.

```
atr x    4         r    n 1  4]  r    n 0  3]
```

$$\begin{bmatrix} & 2 & 3 & 4 \\ 5 & 6 & & 8 \\ 9 & 0 & & 2 \\ 3 & 4 & 5 & 6 \end{bmatrix}$$

`e   Matr x  nte er`

Interchange the second and

.0

■ e F ■at

Here we define our own user-defined function.

```
c■s nv( )    c■s(1/ )
```

■ e V■ d

Pass this function as an argument to **t**.

```
t(c■s nv, 5 2058)
```

.

```
MPOLY( x, ], N)    (x  2-x    3 3  )  2
```

$$x^4 - 2\,y^3\,x^3 \quad y^6 \quad 6\,y)\,x^2 - 6\,y^4\,x \quad 9\,y^2$$

```
e  Mu t var ateP  n  a (
```

```
t(s rt(   2)/ ,    0)
```

$$[le\ \ tH\ \ ndLimit\quad - \ ,ri\ \ htH\ \ ndLimit$$

```
    e    n n(Rec rd( e tHandL   t    n n(OrderedC    et n
                  Ex ress n  nte er,  a  ed ),r   tHandL   t
  n n(OrderedC    et n Ex ress n  nte er,  a  ed )),   )
```

s*TyTdT*   *rTd T*

■ e n var atePu seuxSer es■nr te

1

3.1 . RI

2

2

valuate the series at the v

You can also compute partial derivatives by specifying the order of differentiation.

[illegible]

You can use F, x, and y in expressions.

```
a    F(x   ,      ,     2)    x   (   1)
```

x y z F x z0Td [illegible] 2TfTf6.840Td

$$\Big(2\ z^2 \quad 2\ z)$$

```
c■   ex nte rate(1/(x  2   a),x)
```

$$\frac{\log \frac{x \quad \overline{a} \quad a}{\overline{a}} - \log \frac{x \quad \overline{a} \quad a}{\overline{a}}}{2 \quad \overline{-}}$$

```
                                         ■  e   Ex ress ■n  nte er
```

The following two examples illustrate the limitations of table-based approaches. The two integrands are very similar, but the answer to one of them requires the addition of two new algebraic numbers.

This one is the easy one. The next one looks very similar but the answer is much more complicated.

```
 nte rate(x  3 / (a b x)  (1/3),x)
```

$$20\ b^3\ x^3 - \quad 35 \quad b^2\ x^2 \qquad 62 \quad {}^2\ b\ xT \quad TiT\ TbT$$

conclusively proves that an integral cannot be expressed in terms of elementary functions.

When Axiom returns an integral sign, it has proved that no answer exists as an elementary function.

```
nte rate( ■ (1   s rt(a x   b)) / x,x)
```

$$\int^{x} \frac{\log \quad \overline{b \quad \%}}{\%}\, d\%$$

```
■  e    n ■n(Ex ress ■n  nte er,   )
```

Axiom can handle complicated mixed functions much beyond what you can find in

. If x $\tan t$ and $\tan t/3$ then the following algebraic relation is tr

```
▪ erat▪r
```

$$y$$

```
                                                   ▪  e  Bas c0 erat▪r
```

Here we solve a third order equation with ol nomial coefficients.

```
de     x  3   D(  x, x, 3)   x  2   D(  x, x, 2) - 2   x   D(  x,
x)   2     x   2   x  4
```

$$x^3\, y''' \; x \quad x^2\, y'' \; x \; - 2\, x\, y' \; x \quad 2\, y \; x \quad 2\, x^4$$

```
                                   ▪  e  E uat ▪n Ex ress ▪n  nte er
```

```
s▪ ve(de ,  , x)
```

$$\Big[\; rticul\;\; r \quad \frac{x \quad 10\, x^3 \;\; 20\, x \quad 4}{15\;\; x},$$

$$b \quad i \qquad \Big[2\, x^3 - 3\, x^2$$

$$,\; x_3 - 3\; x_5 -$$

$$,\; x_3 -$$

e V d

Find the real roots of 9 with rational arithmetic, correct to within

```
e ns     x  2 -       ,x  2     x  4 - b  ,    2    - a - b x]
```

yT t o

[

Chapter

Graphics

Figure 4.

Plotting 2D graphs of 1 variable

The general format for drawing a function defined by a formula x is

```
draw( (x), x   a  b, options)
```

where ..b defines the range of x, and where

Plottin 2D aram

Plotting 2D algebraic curv

The general format for drawing a non-singular solution curve given by a polynomial of the form x, y 0 is

```
draw( (x, )   0, x,   , ran e      a  b, c  d], options)
```

where the second and third arguments name the first and second independent variables of . ran e option is always given to designate a bounding rectangular region of the plane x b, c y d. Zero or more additional options as described in 4.0. on page 36 may be given.

third kind of two-dimensional gra

come to a oint cus . lgebraicall fTjnRa25bthisicat4ta o

adaptive	The `adaptive` option turns adaptive plotting on or off. Adaptive plotting uses an

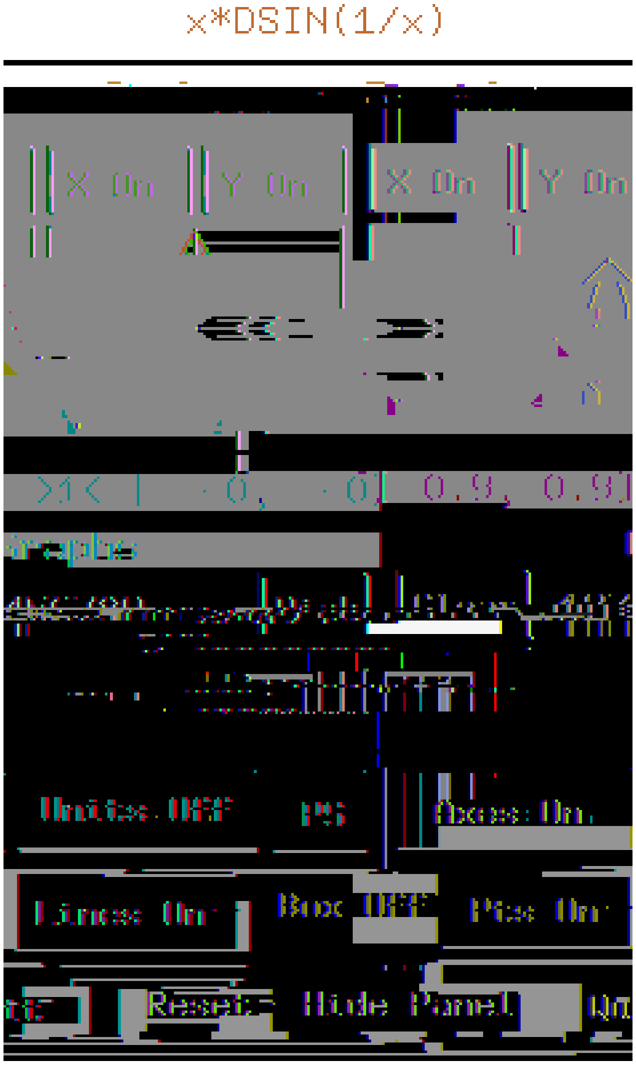

Figure 4.6 Two-dimensional control-panel.

Pick:

axesColorDefault `(` *color* `dark blue` `)) )`
sets or indicates the default color of the axes in a two-dimensional graph viewport.

clipPoint

r iȏn (*v ew ort nte er* 1) *str n* " "))
declares whether graph *int r* is or is not to be displayed with a bounding rectangle.

r t (*v ew or)*

```
8     nt   5,1]$(P nt DFLOA)
```

```
c3      aste    e  w()
```

[Hue eight .0 from

5068 625 0. 2Tf2 6.4820Td[H - 000 P TJ284.920TdT R06824. 920Td4

0 oin

```
          r    n    re eat
           c   nent( , ,  ntC  rDe au t(),  neC  rDe au t(),
(n)  15n, [illegible] n n
```

Plottin 3D function of 2 variabl

The general format for drawing a surface defined a

Plotting 3D Parametric Surfaces

4.0.9 Three-D[redacted]mensional Control-Panel

nce

- **Object:** The **Object** button indicates that the rotation is to occur with respect to the center of volume of the object, independent of the axes' origin position.

Scale: scaling transformation occurs by clicking the mouse

BW converts a color viewport to black and white, or vice-versa. When this button is selected the control-panel and viewport switch to an immutable colormap composed of a range of grey scale patterns or tiles that are used whereveTj8.88sis necessary.

Light takesoj8.88s a control-panel described belo

**i w ** takesoj8.88s anothe [illegible] 560Td of the [illegible] 4.640Td [illegible] 000that [illegible] 80Td [illegible] Tj8.88 [illegible]

the

View Column

The **View Column** button changes the con

tAda tiv 3D (*bool*

vi w cal D fault (*float*

```
-3
```

$$-3$$

```
                                                     ■ e    nte er
```

Here we create a rational number but it looks like the last result.

n domain can be refined to a *subdom in* b a membershi `red cate`. `red cate` is a function that, when a

Pol nomial quareMatrix , om lex Integer

e D an

nother common categor is `F e d`, the class of al2404ofieMds680290. 2Tf5 60Td o TJ 0field0Td 36

. a name for example, `R n`, used to

[redacted]hen might

If the type itself has parentheses around it and we are not in the case of the first example above, then the parentheses can usually be omitted.

```
(2/3)@Fraction(Polynomial Integer)
```

$$\frac{2}{3}$$

```
                                        Type: Fraction Polynomial Integer
```

If the type is used in a declaration and the argument is a single-word type, integer or symbol, then the parentheses can usually be omitted.

```
(d, , )     Complex Polynomial Integer
```

```
                                                          e   V e8 e
```

`?( nte er)`, `Matr x(? (P n a ))`, `S uareMatr x(?, nte er)` it requires a numeric argument and `S uareMatr x(?, ?)` are all invalid.Fhe0Td26.5j-340.32 40Td9re

You can always combine a declaration with an assignment. When you do, it is equivalent to first giving a declaration statement, then giving an

(, ,r) Matr x P■ n■ a ?

■ e V■ d

$$[\ -i\ x \qquad y \quad 4\ i$$

```
                              Type: Matrix Polynomial Complex Integer
```

Note the difference between this and the next example. This is a complex object with polynomial real and imaginary parts.

```
    COMPLEX POLY ?    (x
```

```
    e   Recrd(a    nte er,b   Str n )
```

To access a componen

Records may be nested and the selector names can be shared at different levels.

```
r    Record(a    Record(b    nte er, c    nte er), b    nte er)
```

```
                                                          e   V d
```

The record

rdd

It is possible to create unions like `Union(Integer, PositiveInteger)` but the are difficult to work with because of the overlap in the branch t

. xiom normall converts a result to the target value before assing it to the function. If we left the declaration information out of this function definition then the **a Branch**

3

```
      e     n n(
```

5.5.2 Unions With Selectors

Like records, you can write

$$\left[\;,\; .2, \frac{3}{2}, x^2, \; \texttt{wa} \qquad \right]$$

```
e   L st An
```

hen we ask for the elements, xiom dis la s these t es.

```
u 1
```

```
e   P s t ve nte er
```

ctuall , these objects belong to `An` but xiom automaticall

By default, 3 has the type `PositiveInteger`.

```
3
```

$$\left[\begin{array}{cc} x - \frac{3\ i}{4} & y^2\ z\ \ \frac{1}{2} \\ \frac{3\ i}{7}\ y^4 - x & \frac{60\ \ 9\ i}{5} \end{array}\right]$$

```
e  S uareMatr x(2,P  n  a  Fract  n C   ex  nte er)
```

Interchange the `P  n  a` and the `Fract  n` levels.

```
2    1     S uareMatr x(2,FRAC POLY COMPLEX  N)
```

$$\left[\begin{array}{cc} \frac{4\ x\ \ 3\ i}{4} & \frac{2\ \ \ \ z\ \ 1}{2} \\ \frac{3\ i\ \ \overset{4}{}\ \ 7\ x}{7} & \frac{60\ \ 9\ i}{5} \end{array}\right]$$

```
e  S uareMatr x(2,Fract  n P  n  a  C   ex  nte er)
```

Interchange the `P  n  a` and the `C   ex` levels.

```
3    2     S uareMatr x(2,FRAC COMPLEX POLY  N)
```

$$\left[\begin{array}{cc} \frac{4\ x\ \ 3\ i}{4} & \frac{2\ \ \ \ z\ \ 1}{2\ 278} \end{array}\right.$$

categor

`e Fract n nte er`

It makes sense then that this is a list ofasefnse

`e F at`

Perhaps we a

Sometimes it makes sense, as in this expression, to say choose the operations in this expression so that the final result is Float.

```
(2/3)@Float
```

$$0.6666666666666666666$$

Type: Float

Here we used [illegible]

This says that the operations should be chosen so that the result is a `P■ n■ a` object.

```
((x       % )  2)@(P■  n■  a  C■   ex  nte
                                           (e  % 0(e)(re)■ 8 2 0■d(s)■ 3 8x0■d( s)■
```

$$\begin{bmatrix} \frac{1}{8} & \frac{1}{6} \\ -\frac{1}{4} & \frac{1}{9} \end{bmatrix}$$

`e Mat`

```
categories
        AbelianGroup                              ABELGRP
        AbelianMonoid                             ABELMON
        AbelianMonoidRing                         AMR
        AbelianSemiGroup                          ABELSG
        Aggregate                                 AGG
        Algebra                                   ALGEBRA
        AlgebraicallyClosedField                  ACF
        AlgebraicallyClosedFunctionSpace          ACFS
        ArcHyperbolicFunctionCategory             AHYP
```

For each constructor in a group, the full name and the abbreviation is given. There are other groups in **exposed.lsp** but initially only the constructors in exposure groups “basic” “categories” “naglink” and “anna” are exposed.

As an interactive user of Axiom, you do

This is a polynomial.

```
x x
```

$$2 x$$

```
                                         Type: Polynomial Integer
```

Expose OutputForm.

```
)set expose add constructor OutputForm

   OutputForm is now explicitly exposed in frame G82322
```

This is what we get when OutputForm is automatically available.

```
x x
```

$$x \quad x$$

```
                                         Type: OutputForm
```

Hide OutputForm so we don't run into problems with any later examples

```
)set expose drop constructor OutputForm

   OutputForm is now explicitly hidden in frame G82322
```

Finally, exposure is done on a frame-by-frame basis. *r m* is one of o

operations. The most o

```
 RMA■CA■- Rectan u arMatr xCate ■r &
 RMA■R X  Rectan u arMatr x
 SMA■CA■- S uareMatr xCate ■r &
 S MA■R X S uareMatr x
```

imilarl , if ou wish to see all ackages whose names contain `auss`”, enter this.

```
)w at  ac a e  auss


--------------------- Pac a es ----------------------

Pac a es w t  na es  atc  n   atterns
      auss

 GA SSFAC Gauss anFact■r  at ■nPac a e
```

This commwh hal

tah

)d s a erat n c ex

ere s ne ex sede

Chapter

Using HyperDoc

Figure 6. The HyperDoc root window page

HyperDoc is the gateway to Axiom. It's both an on-line tutorial and an on-line reference manual. It also enables you to use Axiom simply by using the mouse and filling in tem 3silm.is aaerDocHine

Down Arrow croll do

Pa U croll u one

Pa ro ll

[redacted]

The input area [redacted] **Backspace** ke [redacted] the right-arrow $\boxed{\rightarrow}$

The glossary has an input area at its bottom. We review the various kinds of search strings you can

its text When you do, the example line is copied into a new interactive Axiom buffer for this HyperDoc page.

Sometimes one example line cannot be run before you run an earlier one. Don't wo

Chapter 7

Input Files and Output Styles

In this chapter

Turn TeX output on again.

```
)set output tex on
```

The characters used for the matrix brackets above are rather ugly. You get this character set when you issue `)set output characters plain`. This character set should be used when you are running on a machine that does not support the IBM extended ASCII character set. If you are running on an IBM workstation, for example, issue `)set output characters default` to get bet

```
\de \csc {\ at  {\r  csc }\n    ts}

\de \er {\ at  {\r  er }\n    ts}

\de \ a #1#2{
  {{\      \ e t  {#1} \r   t }
   \ ver
   {\ e t  {#2} \r   t  \     }
  }
}
```

7.6 BM S ript ormula ormat

xiom can roduce IBM cri t Formula Format o

ince some versions of FORTRAN have restrictions on the number of lines er statement, xiom breaks long expressions into segments with a maximum of 320 characters 20 lines of 66 characters er segment. If ou want clongj4.920Td8haued seg

e P n a nte er

This ciontntsoleturn 602T3ngeger.09d0tTegentT2j 2.960 singege T j5. 60[l - 000 e TJd Th TJ 2.23.040[l - 000 a

```
R8 S N(EXP
```

Chapter 8

Axiom System Commands

This chapter describes system commands, the command-line facilities used to con

8.2)abbreviation

User Level Required: compiler

$8.5.$

```
)c ear va ue a
)c ear v a
```

This retains whatever declarations the objects had. To remove definitions and values for the specific objects `x,` and , issue

```
)c ear va ue x
)c ear v x
```

To remove the de, clarevTd 2 TJ48.480Td e TJ8.28 thing0Td0r Tj 5.24whil0Td2423

)c e *l N m* a

)c e *dir r s R.12 - o di -1 l di -1m21. il R.1212. d . .1. d l 1..21.2 d co*
co mpil l N m mp.

co mpil l N m . mpmp
co mpil l N m mpmpmpD scriptio mp 1.2. n:il

8.38

```
-O -Fas  -Fal -F s  - ax l  -Mnl-AXL_ _    Obslete -DAx l
```

These options mean

- `-O` erform all optimi ations,
- `-Fas` generate a `as` file,
- `-Fal` generate a `al` file,
- `-F s` generate a

```
)compile matrix.spad
)edit
)compile
```

will call the compiler, edit, and then call the compiler again on the file **matrix.spad**. If you do not specify a *directory*

8.8)display

User Level Required: interpreter

Command Syntax:

```
)display all
)display properties
)display properties all
)display properties
```

```
)s ste  e acs /etc/rc tc
```

calls `e acs`

Some frames are created by the HyperDoc program

```
) e   c ear
```

will display 50

has been issued. Issuing either

```
)set history off
)history )off
```

will discontinue the recording of information.

Whether the facility is disabled or not,

`)reset` will flush the internal list of the most recent workspace calculations so that the data structures may be garbage collected by the underlying Common Lisp system. Like `)history )change`, this option only has real effect when history data is being saved in a file.

`)restore` *[savedHistoryName]* completely clears the environment and restores it to a saved session, if possible. The `)save` option below Tj/R8allows you to save a session to a file with a given name. If you had issued `)history )save jacobi` the command `)history )restore jacobi` would clear the current workspace and load the contents of the named saved session. If no saved session name is specified, the system looks for a file called **last.axh**.

`)save` *savedHistoryName* is used to save a snapshot of the environment in a file. This file is placed in the current working directory. Use `)history )restore` to restore the environment to the state preserved in the file. This option also creates an input file containing all the lines of input since you created the workspace frame (for example, by starting your Axiom session) or last did a `)clear all` or `)clear completely`.

`)show` *n* [`both`] can show previous input lines and o

```
)library )noexpose
```

Command Description:

This command replaces the `)load` system command that was available in Axiom releases

Since this command is only useful for evaluating single expressions, the `)fin` command may

8.19)quit

User Level Required: interpreter

Command Syntax:

```
)quit
)set quit protected | unprotected
```

Command Description:

This command

will read the contents of the file **matrix.input** into Axiom. The ".input" file extension is optional.

This command remembers the previous file you edited, read or compiled. If you do not specify a file name, the previous file will be read.

The `)ifthere` option checks to see whether the

8.22

```
)s ■w POLY  N■ )■ erat ■ns
)s ■w P■  n■  a   nte er
)s ■w P■  n■  a   nte er )■ erat ■ns
```

are among the com

This command is used to create short s non

[redacted]e do not r

`)a r  s` *pattern1* [*pattern2* ...

Command Description:

This command is used

8.29. M K I 26 26

i li gr p y

[Jenks, R.J. and utor, R. . xiom The uti-oc00003i0erTd0eTd6.0 Tj g0Td8d28- 000u33N0Td[e - 0Td86.

Index